Perspecta Labs SecureIO v2.0.4 Security Target

Acumen Security, LLC.

Document Version: 0.8

Table Of Contents

Revision History

Version	Date	Description
0.1	04/06/20	Initial Draft
0.2	09/04/20	Addressed Comments
0.3	09/09/20	Addressed QA Team Comments
0.4	09/15/20	Addressed TL comments
0.5	09/22/20	Addressed Vendor’s comments
0.6	01/15/2021	Addressed ECR check-in comments
0.7	06/07/2021	Addressed latest TDs and added X.509 claims
0.8	06/30/2021	Addressed ECR comments

1 Security Target Introduction

1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Category	Identifier
ST Title	Perspecta Labs SecureIO v2.0.4 Security Target
ST Version	0.8
ST Date	June 30, 2021
ST Author	Acumen Security, LLC.
TOE Identifier	Perspecta Labs SecureIO v2.0.4
TOE Software Version	2.0.4
TOE Developer	Perspecta Labs
Key Words	TLS Proxy

Table 1 TOE/ST Identification

1.2 TOE Overview

The SecureIO application provides a secure communication channel for Android applications by transmitting and receiving network traffic over a secure TLS channel. The traffic will be protected in transit using TLS between the Android device and a TLS server. Figure 1 below provides an overview and indicates the TOE boundary with a red-dotted line.

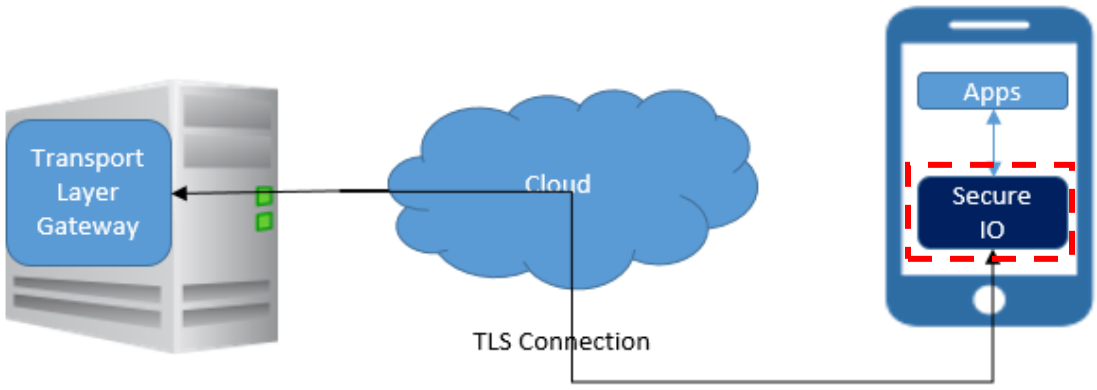

Figure 1 SecureIO Overview

The functionality of the SecureIO service is limited to (i) establishing and shutting down a TLS connection to the Transport Layer Gateway (TLG); (ii) sending and receiving messages to and from the TLG on behalf of Android apps via the TLS connection.

1.3 TOE Architecture

1.3.1 Physical Boundaries

The TOE is a software application that resides entirely on its Android-based mobile platform.

1.3.1.1 Software Requirements

The TOE runs on Android versions 8.0, 9.0, and 10.0. All sub-versions of 8.0 (e.g. 8.1.0), 9.0 and 10.0 are supported.

1.3.2 Security Functions provided by the TOE

The TOE provides the security functionality required by Protection Profile for Application Software Version 1.3 [SWAPP].

1.3.2.1 Cryptographic Support

The TOE relies on underlying cryptographic functionality provided by the platform for all of its cryptographic operations.

1.3.2.2 User Data Protection

The TOE is a TLS proxy that encrypts data sent by other applications on its host platform.

1.3.2.3 Security Management

The TOE does not come with any default credentials. It identifies itself to the TLS gateway that it connects to using a certificate and private key. These are provisioned onto the TOE by an administrator or end user.

1.3.2.4 Privacy

The TOE itself does not contain or transmit any PII. It functions as a TLS proxy over which other applications on the platform may transmit whatever data they wish.

1.3.2.5 Protection of the TSF

The TOE employs several mechanisms to ensure that it is secure on the host platform. Only documented platform APIs are used by the TOE. The TOE never allocates memory with both write and execute permission. Evaluated platform functionality is used to verify the TOE version and perform updates, and no third-party libraries are used.

1.3.2.6 Trusted Path/Channels

TLS is used to protect all data transmitted to and from the TOE.

1.3.2.7 Identification and Authentication

Certificate validation and certificate authentication performed by the TOE as part of TLS, in accordance with RFC 5280.

1.3.3 TOE Documentation

- [ST] Perspecta Labs SecureIO v2.0.4 Security Target, Version 0.7
- [AGD] Perspecta Labs SecureIO User Manual, Version 2.0.4

1.3.4 Other References

Protection Profile for Application Software, version 1.3, dated, 01 March 2019 [SWAPP].

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 3 extended

2.2 Protection Profile Conformance

This TOE is conformant to:

- Protection Profile for Application Software, Version 1.3, dated 01 March 2019 [SWAPP]

2.3 Conformance Rationale

This Security Target provides exact conformance to Version 1.3 of the Protection Profile for Application Software. The security problem definition, security objectives, and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to [SWAPP] have been addressed. The following table identifies all applicable TDs:

Identifier	Applicable	Exclusion Rationale (if applicable)
TD0587 – X.509 SFR Applicability in App PP	Yes	
TD0582 – PP-Configuration for Application Software and Virtual Private Network (VPN) Clients now allowed	Yes	
TD0561 – Signature verification update	Yes	
TD0554 – iOS/iPadOS/Android AppSW Virus Scan	Yes	
TD0548 – Integrity for installation tests in AppSW PP 1.3	Yes	
TD0544 – Alternative testing methods for FPT_AEX_EXT.1.1	Yes	
TD0543 – FMT_MEC_EXT.1 evaluation activity update	No	This TD only applies to Windows platforms. The TOE runs on Android.
TD0540 – Expanded AES Modes in FCS_COP	No	FCS_COP SFRs are not claimed by the TOE.
TD0519 – Linux symbolic links and FMT_CFG_EXT.1	No	This TD only applies to Linux platform. The TOE runs on Android.
TD0515 – Use Android APK manifest in test	Yes	
TD0510 – Obtaining random bytes for iOS/macOS	No	This TD only applies to iOS platforms. The TOE runs on Android.
TD0498 – Application Software PP Security Objectives and Requirements Rationale	Yes	
TD0495 – FIA_X509_EXT.1.2 Test Clarification	Yes	
TD0473 – Support for Client or Server TOEs in FCS_HTTPS_EXT	No	HTTPS protocol is not implemented by the TOE.

TD0465 – Configuration Storage for .NET Apps	No	This TD only applies to Windows platforms. The TOE runs on Android.
TD0445 – User Modifiable File Definition	Yes	
TD0437 – Supported Configuration Mechanism	Yes	
TD0435 – Alternative to SELinux for FPT_AEX_EXT.1.3	No	This TD only applies to Linux platforms. The TOE runs on Android.
TD0434 – Windows Desktop Applications Test	No	This TD only applies to Windows platforms. The TOE runs on Android.
TD0427 – Reliable Time Source	Yes	
TD0416 – Correction to FCS_RBG_EXT.1 Test Activity	Yes	

Table 2 SWAPP Technical Decisions

3 Security Problem Definition

The security problem definition has been taken from [SWAPP] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1 Threats

The following threats are drawn directly from the [SWAPP].

ID	Threat
T.NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.
T.NETWORK_EAVESDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.
T.LOCAL_ATTACK	An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
T.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.

Table 3 Threats

3.2 Assumptions

The following assumptions are drawn directly from the SWAPP.

ID	Assumption
A.PLATFORM	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
A.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

Table 4 Assumptions

3.3 Organizational Security Policies

There are no OSPs for the application.

4 Security Objectives

The security objectives have been taken from [SWAPP] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The following security objectives for the TOE were drawn directly from the SWAPP.

ID	TOE Objective
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options. Addressed by: FDP_DEC_EXT.1, FMT_CFG_EXT.1, FPT_AEX_EXT.1, FPT_TUD_EXT.1
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs. Addressed by: FMT_MEC_EXT.1, FPT_API_EXT.1, FPT_LIB_EXT.1, FPT_TUD_EXT.2
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII. Addressed by: FMT_SMF.1, FPT_IDV_EXT.1, FPT_TUD_EXT.1, FPR_ANO_EXT.1
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data. Addressed by: FDP_DAR_EXT.1, FCS_STO_EXT.1, FCS_RBG_EXT.1
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application. Addressed by: FTP_DIT_EXT.1, FCS_RBG_EXT.1, FCS_CKM_EXT.1, FDP_NET_EXT.1

Table 5 Objectives for the TOE

4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

ID	Objective for the Operation Environment
OE.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
OE.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.
OE.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

Table 6 Objectives for the environment

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5 and all international interpretations.

Requirement	Requirement Description
Mandatory SFRs	
FCS_RBG_EXT.1	Random Bit Generation Services
FCS_CKM_EXT.1	Cryptographic Key Generation Services
FCS_STO_EXT.1	Storage of Credentials
FDP_DEC_EXT.1	Access to Platform Resources
FDP_NET_EXT.1	Network Communications
FDP_DAR_EXT.1	Encryption Of Sensitive Application Data
FMT_MEC_EXT.1	Supported Configuration Mechanism
FMT_CFG_EXT.1	Secure by Default Configuration
FMT_SMF.1	Specification of Management Functions
FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
FPT_API_EXT.1	Use of Supported Services and APIs
FPT_AEX_EXT.1	Anti-Exploitation Capabilities
FPT_TUD_EXT.1	Integrity for Installation and Update
FPT_LIB_EXT.1	Use of Third Party Libraries
FPT_IDV_EXT.1	Software Identification and Versions
FTP_DIT_EXT.1	Protection of Data in Transit
Optional, Selection-Based and Objective SFRs	
FPT_TUD_EXT.2	Integrity for Installation and Update
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication

Table 7 SFRs

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations matches the formatting specified within the PP.

5.2 Security Functional requirements

5.2.1 Cryptographic Support (FCS)

FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [

- use no DRBG functionality

] for its cryptographic operations.

FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1

The application shall [

- generate no asymmetric cryptographic keys

].

FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

The application shall [

- invoke the functionality provided by the platform to securely store *X.509 certificates*

] to non-volatile memory.

5.2.2 User Data Protection (FDP)

FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1

The application shall restrict its access to [

- network connectivity.

].

FDP_DEC_EXT.1.2

The application shall restrict its access to [

- no sensitive information repositories.

].

FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1

The application shall restrict network communication to [

- user-initiated communication for *secure tunnel establishment*.

].

FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1

The application shall [

- not store any sensitive data

] in non-volatile memory.

5.2.3 Security Management (FMT)

FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1[1]

The application shall [

- invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.]

FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1

The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2

The application shall be configured by default with file permissions which protect the application binaries and data files from modification by normal unprivileged users.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions [

- no management functions.

].

5.2.4 Privacy (FPR)

FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1

The application shall [

- not transmit PII over a network.

].

5.2.5 Protection of TSF (FPT)

FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1

The application shall only use documented platform APIs.

FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1

[1] TD0437 applied.

The application shall not request to map memory at an explicit address except for [*no exceptions*].

FPT_AEX_EXT.1.2

The application shall [

- not allocate any memory region with both write and execute permissions.

].

FPT_AEX_EXT.1.3

The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4

The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5

The application shall be built with stack-based buffer overflow protection enabled.

FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1

The application shall [

- leverage the platform

] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2

The application shall [

- provide the ability

] to query the current version of the application software.

FPT_TUD_EXT.1.3

The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.4[2]

Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5

The application is distributed [

- as an additional software package to the platform OS

].

[2] TD0561 applied.

FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1

The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.2.2

The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.2.3[3]

The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1

The application shall be packaged with only [*no third-party libraries*].

FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1

The application shall be versioned with [

- SWID tags that comply with minimum requirements from ISO/IEC 19770-2:2015

].

5.2.6 Trusted Path/Channel (FTP)[4]

FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1

The application shall [

- invoke platform-provided functionality to encrypt all transmitted data with [TLS]

] between itself and another trusted IT product.

5.2.7 Identification and Authentication (FIA)[5]

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

The application shall [invoke platform-provided functionality] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage

[3] TD0561 applied.
[4] TD0587 applied.
[5] TD0587 applied.

field

- The application shall validate the revocation status of the certificate using [OCSP as specified in RFC 6960, CRL as specified in RFC 5759].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

FIA_X509_EXT.1.2

The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS].

FIA_X509_EXT.2.2

When the application cannot establish a connection to determine the validity of a certificate, the application shall [not accept the certificate].

5.3 TOE SFR Dependencies Rationale for SFRs

The Protection Profile for Application Software contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the Protection Profile for Application Software which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documentation	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Life-Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
	ALC_TSU_EXT.1	Timely Security Updates
Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition

Assurance Class	Components	Components Description
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

Table 8 Security Assurance Requirements

5.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

5.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Perspecta Labs to satisfy the assurance requirements. The table below lists the details.

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ALC_TSU_EXT.1	Perspecta Labs uses a systematic method for identifying and providing security relevant updates to the TOEs users via its support infrastructure.
ATE_IND.1	Perspecta Labs will provide the TOE for testing.
AVA_VAN.1	Perspecta Labs will provide the TOE for testing.

Table 9 TOE Security Assurance Measures

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Requirement	Rationale
FCS_RBG_EXT.1	The TOE does not use DRBG functionality for its cryptographic operations. Due to its leveraging of platform cryptographic functionality there are no TOE functions covered by ST SFRs that directly use random numbers provided by the platform for cryptographic operations. All random numbers that are used as part of the platform protocols are invoked directly by the platform protocols and not by the TOE.
FCS_CKM_EXT.1	The application does not generate any asymmetric cryptographic keys.
FCS_STO_EXT.1	Digital certificates (and the keys associated with the digital certificates), which are the secure credentials used for connection authorization by the TOE, are stored within the Android key Chain on the platform. When needed, the user selects the credentials to use from the platform itself. To properly verify OCSP, the TOE access instances of the java java.security.KeyStore for passing and processing public certificates.
FDP_DAR_EXT.1	During operation of the TOE, no sensitive data is stored in non-volatile memory. It is not possible for the TOE to store such data because it never receives it.
FDP_DEC_EXT.1	During operation of the TOE, access to the underlying platform is limited to use of network connectivity hardware for establishment of secure communication channels. No sensitive information repositories are accessible.
FDP_NET_EXT.1	During regular operation of the TOE, secure TLS sessions may be established to provide secure channels for communications. These interactions are performed based on the following events: • Pressing the “Connect” button [User guide, Section 4 Operation]
FMT_CFG_EXT.1	The TOE does not come with any default credentials. It identifies itself to the TLS gateway that it connects to using a certificate and private key. These are provisioned onto the TOE by an administrator by uploading the necessary certificates to the Android platform.
FMT_MEC_EXT.1 FMT_SMF.1	The TOE maintains a restricted configuration with no management functions being performed by users. Administrative users are the only types of users for this TOE.
FPR_ANO_EXT.1	The TOE does not transmit any PII over the network.
FPT_AEX_EXT.1	Because the TOE is a pure Java application it is not necessary to enable ASLR as Java natively checks array bounds. TOE is developed in Java using its built-in buffer overflow protection, which is enabled by default in the Android Studio/gradle scripts used to build the software.

FPT_API_EXT.1	The TOE leverages the following platform APIs: • android.security.KeyChain • java.io.IOException; • java.io.InputStream; • java.net.InetAddress; • java.net.InetSocketAddress; • java.net.SocketException; • java.security.KeyStore; • java.security.PrivateKey; • java.security.PublicKey; • java.security.Signature; • java.security.cert.Certificate; • java.security.cert.X509Certificate; • javax.net.ssl.HostnameVerifier; • javax.net.ssl.KeyManager; • javax.net.ssl.KeyManagerFactory; • javax.net.ssl.SSLHandshakeException; • javax.net.ssl.SSLPeerUnverifiedException; • javax.net.ssl.SSLSession; • javax.net.ssl.SSLSocket; • javax.net.ssl.SSLSocketFactory; • javax.net.ssl.TrustManager; • javax.net.ssl.TrustManagerFactory; • javax.net.ssl.X509TrustManager;
FPT_LIB_EXT.1	The TOE does not come packaged with any third-party libraries.
FPT_TUD_EXT.1 ALC_TSU_EXT.1	The TOE leverages the underlying platform to check for updates and patches to the application software. All updates are packaged in the Android Application Package (APK) format and distributed as an additional software package to the platform OS. All updates are digitally signed to ensure they are provided by Perspecta Labs, which is the only authorized source for software updates. In the event that any security vulnerability applies to SecureIO, Perspecta Labs will deliver an update within 30 days. Users of the SecureIO app should report any security related issues to the Perspecta Labs support team at secureio-support@perspectalabs.com. Customers can use this email address to request ability to send encrypted email, and Perspecta Labs support personnel will respond with a digitally signed email. Customers can then respond to the email with a digitally encrypted email, including any attachments.
FPT_TUD_EXT.2	The updates are packaged in the Android Application Package (APK) format. The updates are digitally signed by Perspecta Labs as publisher. The platform verifies the authenticity of the update by cryptographically verifying that the digital signature of the offered update matches the digital signature of the currently installed app.
FPT_IDV_EXT.1	The Application is versioned with SWID tags that comply with minimum requirements from ISO/IEC 19770-2:2015.

FTP_DIT_EXT.1	All communication sent between the TOE and any external IT entity is encrypted to protect all transmitted data. This communication is performed over TLS.
	
	Perspecta Labs SecureIO uses the Standard JAVA SSLContext in android to get access to the built in Android System libraries. This results in a socket Factory that SecureIO uses to create the TLS wrapped SSLSocket.
	
	SSLContext sslContext = SSLContext.getInstance("TLS");
	SSLSocketFactory factory = sslContext.getSocketFactory();
	
	In addition, prior to the handshake being sent, SecureIO uses the setEnabledProtocols method on the resulting SSLSocket to force TLS 1.2 only, as well as setEnabledCipherSuites to limit the cipher suites to the ones listed on the connection Profile in the SecureIO application.
	
	The current list of allowable cipher suites is shown below:
	TLS_RSA_WITH_AES_128_CBC_SHA
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
	TLS_RSA_WITH_AES_128_CBC_SHA256
	TLS_RSA_WITH_AES_256_CBC_SHA256
	
	private static final String[] PROTOCOLS = {"TLSv1.2"};
	socket.setEnabledProtocols(PROTOCOLS);
	socket.setEnabledCipherSuites(profile.getCipherSuitesArray());
	socket.startHandshake();
	
	Then all application network traffic is sent/received using the SSLSocket.

FIA_X509_EXT.1 FIA_X509_EXT.2	Certificate validation and certificate path validation performed by the TOE is conformant with RFC 5280. The TOE is configured with a single certificate which is used for all communication. The TOE performs certificate validation and follows the certificate path validation algorithm as follows: The TOE supports chains of length four or greater. Certificates received as part of TLS connections are checked for a valid path up to the certificate authority roots (which must have the X509v3 Basic Constraint CA: True). The notBefore and notAfter dates included in certificates will be checked to be before and after the current time respectively. The TOE validates that the certificate path must terminate with a trusted CA certificate. The TOE validates that any CA certificate includes caSigning purpose in the key usage field. The TOE validates the extendedKeyUsage (EKU) field for the Server certificates presented for TLS to have the

	Server Authentication purpose and OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose in the EKU field. Validity checks are performed by the TOE, using functionality implemented in the TOE. For certificates to successfully validate, the certificate cannot be revoked. Certificate revocation is determined using either a CRL check or OCSP. In addition to the revocation check, the certificate must have a valid basicConstraints extension and extendedKeyUsage field. If for any reason the TOE is unable to determine the validity of a certificate, the certificate will not be accepted.

Table 10 TOE Summary Specification SFR Description

7 Acronym Table

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
APK	Android Application Package
CC	Common Criteria
CMC	Certificate Management over CMS
CMS	Cryptographic Message Syntax
DRBG	Deterministic Random Bit Generator
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
NIAP	Nation Information Assurance Partnership
OS	Operating System
PII	Personally Identifiable Information
PKG	Package file
PP	Protection Profile
RBG	Random Bit Generator
RSA	Rivest, Shamir, & Adleman
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
ST	Security Target
SWID	Software Identification
TD	Technical Decision
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Functions

Acronym	Definition
TSS	TOE Summary Specification

Table 11 Acronyms